DECLARATION OF PROFESSOR MICHEL DUBOIS REGARDING CLAIM CONSTRUCTION FOR U.S. PATENT NO. 5,749,095

I, Michel Dubois, hereby declare:

The statements contained in this Declaration are true and correct. If called as a witness, I would testify thereto under oath.

I have been retained by Plaintiff and Counterclaim Defendant NetApp, Inc. to offer opinions regarding certain claim terms in U.S. Patent No. 5,749,095 ("'095 Patent").

I.

QUALIFICATIONS

I have been a Research Engineer at the Central Research Laboratory of Thomson-CSF in France, and I am currently Professor of Computer Engineering at the University of Southern California, a position I have held for 24 years. I am an IEEE Fellow and an ACM Fellow. My additional qualifications to render an expert opinion in the matter are set forth in my Curriculum Vitae, which is attached as Exhibit A. My C.V. also includes a list of all publications authored in the last 28 years.

II.

STATEMENT OF OPINIONS – '095 PATENT

A summary of my opinions regarding certain claim terms in the '095 patent is set forth below. I reserve the right to modify or supplement my opinions as appropriate.

A. ORDINARY SKILL IN THE ART

One of ordinary skill in the art relevant to the '095 patent in 1996 would generally have the following education and experience: a bachelor's or master's degree in Electrical Engineering or Computer Science, or equivalent experience, and several years experience in working with multiprocessing computer systems.

My opinion is based upon my personal knowledge and experience and my consideration of the following factors: (1) the levels of education and experience of persons of skill working in the field; (2) the types of problems encountered in the art; (3) the prior art patents and publications; (4) the activities of others; (5) prior art solutions to the problems encountered by

the inventor; (6) the sophistication of the technology; and (7) the rapidity with which innovations are made.

B. BACKGROUND

One of ordinary skill in the art reading the '095 patent in 1999 would have understood that it is directed generally to a technique for improving the performance of a multiprocessing computer system by providing a mechanism for carrying out special "fast write" operations that the processor can complete more quickly than traditional write operations. Notably, the alleged invention of the '095 patent is directed to the use of such "fast write" operations in *multiprocessing* systems, where issues of "coherency" can arise.

These "coherency" issues can be connected to the fact that multiprocessing systems may maintain multiple copies of a single piece of shared data – one copy for each processor, for example. More specifically, each processor in a multiprocessing system typically has associated with it a small memory (a "cache"), which the processor can access rapidly and can use to store frequently needed data. A processor can see data in its own cache, but not the data in the caches of other processors. To illustrate this coherency problem, consider a situation where more than one cache is holding the same piece of data. In this situation, a problem may arise if one processor should choose to modify the copy of the data it is holding in its cache. Indeed, should a processor modify a copy of data only in its own cache, then different processors may no longer have a uniform view of the data: the processor that has modified the data will view the data as having the new value, while the processors that have not modified the data will view the data as having the old value. To remedy this "coherency" problem, coherency activity must be carried out. Specifically, the processor modifying the data must do one of two things. It must either (1) inform the other processors that the copy of the data they are holding is invalid, or (2) provide an updated copy of the data to the other processors.

The '095 patent claims special "fast write" operations that may, for instance, be completed "with respect to a processor" prior to the completion of "coherency operations." Representative Claim 11, for instance, includes the following limitations:

> wherein said system interface is configured to complete said write operation with respect to said processor prior to completing said coherency operation if said write operation includes a specific predefined encoding, and wherein said system interface is further configured to inhibit completion of said write operation with respect to said processor until completion of said coherency operation if said write operation includes a different encoding

Recognizing that under certain circumstances, write operations can be "completed with respect to a processor" prior to the completion of coherency operations without resulting in incorrect program execution, a computer programmer may choose to utilize the specially encoded "fast write" operations to obtain a performance advantage.

As the patent explains, the "present invention" contemplates a "system interface" that carries out a number of core functions pertinent to the execution of write operations, including (1) receiving a write operation, (2) performing coherency operations in response to a write operation, (3) completing "fast write" operations with respect to a processor, and (4) inhibiting the completion of traditional write operations pending the completion of coherency activity. *See* '095 patent at 5:53-65. The specification explains how the "system interface" is used for carrying out write operations as follows. First, a "system interface" detects a "fast write" operation by its special encoding. *See id.* at 27:15-16. Upon detecting the "fast write" operation, the "system interface" declines to assert a special signal, called the "ignore signal," which is normally asserted to inhibit further processing of the operation pending the completion of coherency activity. *See id.* at 12:26-43. Instead, the write data is immediately transferred from the initiating processor to a storage queue within the "system interface" prior to the completion of the requisite coherency activity. *See id.* at 27:16-25. As a result, the specification explains, the write operation may "appear to the issuing processor 16 to complete" or, put another way, be "completed in the local node." *See id.* at 27:25-28; *id.* at 27:40-47.

In addition, the specification explains the special feature of the "system interface" that allows this type of "completion" to yield tangible performance benefits. Specifically, as compared to a storage buffer on a processor itself, the "system interface" storage queue is much larger. *See id.* at 29:21-55. Thus, rather than storing a small number of transactions in the processor pending the completion of "coherency operations," which may impede subsequent

operations that require the same processor resources, a large number of transactions can be stored in the off-processor "system interface." *See id.* According to the patent, this process has the benefit that "[p]rocessor resources are freed more rapidly than if the coherency state is acquired prior to receiving the data from the processor." *Id.* at 27:29-31; *see also id.* at 30:63-31:13 (explaining that when the "ignore signal" is not asserted, data is transferred on the bus connected to the "system interface" so that "processor 16 resources used to store and perform the fast write stream transaction are freed rapidly, allowing the resources to be used for subsequent transactions . . . .").

C. THE DISPUTED TERMS

1. "Completing [a] Write Operation Within [a] Local Processing Node" / Completing [a] Write Operation With Respect to [a] Processor (Claims 1, 11, and 17)[1]

One of ordinary skill in the art at the time the '095 patent's application was filed in 1996 would have understood that the claim terms "completing a write operation with respect to said processor" or "completing a write operation within said local processing node" in the claims of the '095 patent means "transferring the write data from an initiating processor to a system interface." In both cases the word "said" is used to make clear that the processor being referred to is the processor issuing the write.

BASIS AND REASONS:

The issue posed by these terms is what the specific "finish line" is for a write operation to be considered "complete" "with respect to said processor" or "within said local processing node." The specification explains what this "finish line" is and how it corresponds to the transfer of write data to the "system interface." As a first example of the specification explaining this to be the case, consider Col. 27:15-64, which steps through the process of carrying out the "fast write" operations to which the claims are directed. There the specification explains

[1] References to claim numbers in this declaration are intended as reference aids, but do not limit the scope of my opinions. To the extent the terms defined in this report are found in claims other than those listed, the same terms in all claims share the same definition unless otherwise noted.

that the "system interface" first recognizes such operations as special operations. *See id.* at 27:15-16. Then, the passage explains that the next step is the transfer of the write data to the "system interface":

> Instead of first acquiring a coherency state for the affected coherency unit consistent with performing a write operation and then subsequently transferring the data from the initiating processor, *system interface 24 allows transfer of the data to system interface 24 prior to completing the requisite coherency operation*.

Id. at 27:16-23. In other words, in contrast to other write operations, "fast write" operations allow for the transfer of data to the "system interface" prior to the completion of the "requisite coherency operation." The specification goes on to explain how this corresponds to the recited "completing":

> The write operation . . . may thereby *appear to the issuing processor to be complete* before the obtaining of the write permission. . . .".

Id. at 27:25-27. In clarifying this, the specification further explains as follows:

> [T]he "fast write" write operation is effectively completed outside of the global ordering of computer system 10 since *the operation is completed in the local node prior to acquiring a coherency state* consistent with performing a write operation.

Id. at 27:43-47. Thus, the specification equates completion "with respect to said processor" and completion "within said local processing node" to the transfer of the write data to the "system interface."

The specification explains how this transfer provides specific benefits. Specifically, the specification explains that "[p]rocessor resources are freed more rapidly than if the coherency state is acquired prior to receiving the data from the processor." *Id.* at 27:29-31. These benefits are set forth repeatedly in the specification. For instance, the specification explains that "[p]rocessor resources are freed upon transmission of the write operation and corresponding data to the system interface, before an appropriate coherency state is acquired by the node containing the processor." *See id.* at 31:36-53. Likewise, the specification explains that "transfer of the data from the initiating processor, [frees] local node resources more quickly than if the same NUMA write transaction were performed using a non-fast write encoding." *See id.* at 28:15-33. Accordingly, it appears that the benefits of the alleged invention of the '095 patent are directly connected to the rapid transfer of data to the "system interface." As yet another example,

the specification explains that the transfer of data upon the data bus connected to the system interface results in "processor 16 resources used to store and perform the fast write stream transaction [being] freed rapidly, allowing the resources to be used for subsequent transactions such as another write stream operation." *Id.* at 30:63-31:13; *see also id.* at Fig. 2 (depicting the referred to data bus as being connected to the system interface). As one of skill in the art, I believe it makes sense to understand the claims in terms of the mechanism that actually provides the benefits of the invention.

The specification confirms my understanding. In addition to the instance described above, other portions of the specification also describe "completion" in terms of transfer to the "system interface." For instance, the specification explicitly explains what it means to "complete a write operation with respect to a processor" as follows: "During step 322, the data is received and stored by system interface 24. *The write operation is thereby complete with respect to the initiating processor* 16." *Id.* at 28:46-52.

As yet another example, the Abstract explains that upon detecting a fast write operation, "the data is transferred to the system interface from the processor" and that the "coherency activity employed to acquire the proper coherency state is initiated subsequent to or in parallel with the receipt of data from the processor." *Id.* at Abstract. The Abstract goes on to explain that, as a result, the "fast write operations are *performed* prior to acquiring write permission to the coherency unit." *Id.* at Abstract; *see also id.* at 5:17-18 (offering a nearly identical description). In offering a similar description later in the specification, the '095 patent explains that this results in the processor being freed up to perform other tasks. *Id.* at 5:7-16. Stating that the write operations are "performed," and connecting this to the benefits of the invention, these descriptions again demonstrate that the relevant "completion" in the '095 patent corresponds to the transfer of the write data to the "system interface."

Interestingly, the specification also describes special characteristics of the system interface, and again connects its use to completion of a write operation. According to the patent, the "system interface" is "much larger than the buffers included within processors," including a

larger number of storage locations in a "data queue." The specification explains that "[d]ue to the larger number of storage locations . . . a large number of fast write stream operations may be queued therein. *Id.* at 29:45-52. As a result, the patent explains, "*the fast write stream transactions are completed from processors* 16 . . . [and] processors 16 may continue with other operations while system interface 24 completes the write stream operations." *Id.* at 29:45-52. Having specifically described the special nature of the "system interface" and the benefits that are derived from its use, the patent indicates that any invention in the '095 patent resides within the special "system interface." Indeed, this passage again specifically equates write operations being "completed from processors," which is the subject of the claims to the transfer of write data to the "system interface."

Although I am informed that Sun's primary position is that these terms are "clear on their face," these phrases simply do not have a well understood meaning in the art. The word "complete," for instance, is a generic term with no agreed upon definition and should thus be defined within the context of the patent. To the extent Sun suggests a definition for the key claim terms that use this word, Sun fails to pin-point the actual step in the execution of a write operation when the write is "complete with respect to said local processing node" or "complete with respect to said processor." Instead. Sun's definitions simply describe the way a write operation should be treated during its lifetime in the system. It is obvious that during its lifetime the write must return its value to dependent reads and that it "is or will be coherent" at one point. But, this still fails to bring any clarity to the meaning of the terms "completing a write operation with respect to said processor" or "completing a write operation within said local processing node."

Sun fails to afford key portions of the claim terms adequate meaning, if any meaning at all. The claims do not simply include the term "completing a write operation." Rather, they include the more detailed terms "completing a write operation with respect to said processor" and "completing a write operation within said local processing node." For the term "completing a write operation within a local processing node," Sun's construction repeats the language "within a local processing node," thus failing to explain or delve into the meaning of

this portion of the claim term. Similarly, for the claim term "completing a write operation with respect to said processor," Sun simply substitutes the phrase "with respect to a processor" with the phrase "by a processor," which, in my opinion, again fails to adequately set forth the meaning of the full claim term.

I conclude that one of ordinary skill in the art would understand that "completing a write operation with respect to said processor" or "completing a write operation within said local processing node" as used in the claims of the '095 patent means "transferring the write data from an initiating processor to a system interface."

III.

MATERIALS REVIEWED

A list of the materials that I reviewed in preparing this report is attached as Exhibit B.

IV.

COMPENSATION

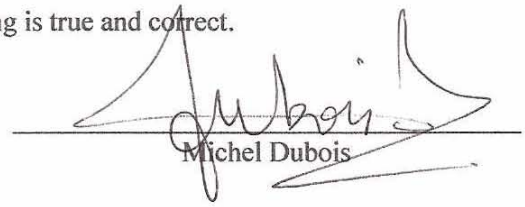
My compensation for consulting on this matter is $350 per hour. My compensation does not depend on the outcome of this dispute.

V.

PREVIOUS TESTIMONY

I have not previously offered deposition or trial testimony.

I declare under penalty of perjury under the laws of the United States of America and the State of California that the foregoing is true and correct.



Michel Dubois